PCT/IB05/01150
04.05.05

REC'D 04 MAY 2005
WIPO PCT

PA 1287161

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

February 23, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: *60/618,688*
FILING DATE: *October 14, 2004*

PCT/IB05/01150

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

P. R. Grant

P. R. GRANT
Certifying Officer

PTO/SB/16 (09-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Express Mail Label No. EV 517424149

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
BISWAROOP	MUKHERJEE	OTTAWA, ONTARIO, CANADA

Additional inventors are being named on the ONE *separately numbered sheets attached hereto*

TITLE OF THE INVENTION (500 characters max):

PAIRWISE INDEPENDENT SCHEDULING FOR NODES WITH A SINGLE PARAMETERIZABLE RADIO IN A WIRELESS NETWORK

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

☑ The address corresponding to Customer Number: 00626

OR

☐ Firm or Individual Name

Address

City	State	Zip
Country	Telephone	Fax

ENCLOSED APPLICATION PARTS *(check all that apply)*

☑ Specification *Number of Pages* 17

☑ Drawing(s) *Number of Sheets* 4

☐ Application Data Sheet. See 37 CFR 1.76

☐ CD(s), Number of CDs

☐ Other (specify)

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check or money order is enclosed to cover the filing fees.

☐ Payment by credit card. Form PTO-2038 is attached.

☑ The Director is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 500210
A duplicative copy of this form is enclosed for fee processing.

FILING FEE Amount ($): 160.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☐ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are:

SIGNATURE *John D. Crane* Date October 14, 2004

TYPED or PRINTED NAME John Crane REGISTRATION NO. 25231 *(if appropriate)*

TELEPHONE 972-685-8442 Docket Number: 16892ROUS02P

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

PROVISIONAL APPLICATION COVER SHEET
Additional Page

PTO/SB/16 (09-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

First Named Inventor	BISWAROOP MUKHERJEE	Docket Number 16892ROUS02P

INVENTOR(S)/APPLICANT(S)

Given Name (first and middle [if any])	Family or Surname	Residence (City and either State or Foreign Country)
BILL	GAGE	STITTSVILLE, ONTARIO, CANADA
JERRY	CHOW	KANATA, ONTARIO, CANADA

Number ONE of ONE

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

16892ROUS02P

UNITED STATES PROVISIONAL PATENT APPLICATION

Biswaroop Mukherjee
Bill Gage
Jerry Chow
INVENTORS

PAIRWISE INDEPENDENT SCHEDULING FOR NODES WITH A SINGLE PARAMETERIZABLE RADIO IN A WIRELESS NETWORK

Docket No. 7000-365-P

PAIRWISE INDEPENDENT SCHEDULING FOR NODES WITH A SINGLE PARAMETERIZABLE RADIO IN A WIRELESS NETWORK

Background of the Invention

[0001] A network where each node shares a single communication resource to exchange data between its (possibly many) neighbors needs to determine when and how it would distribute such resources. If the neighbours contend for time (e.g. on a node's CPU or shared radio(s)) then the problem is of determining the time a node should devote to each of its neighbours so that maximum throughput is achieved. The case of particular interest here is that of a radio that needs to be multiplexed over the many neighbours, where, in order to obtain maximum throughput we need to maximize useful time that neighbours have to communicate with each other while avoiding collisions. This determination of timing and sequence of communication events is referred to here as the scheduling problem. In other words the problem is for each node to create a schedule that determines when and for how long to exchange data with its neighbours avoiding collision. As an example consider Figure 1 which shows a network where each node has a directional radio that it multiplexes over its various links.

Qualifiers

[0002] Widespread nodes: One reason to create a wireless mesh network is to provide widespread coverage by means of wireless nodes that form the backhaul infrastructure. This is an attempt to extend the model of the model of wired node access points where backhaul is over the wired network. In order to be successful the nodes are expected to be spread out in space, with the obvious requirement that at least one node connected to the existing mesh be able to communicate with a new outlying node. So the scheduling solution cannot rely on all the nodes being able to "hear" schedule announcements from one source. See for example Figure 1 where nodes C and I extend the reach of the

wired access points A and B. As will be apparent to one of skill in the art, the signaling from A or B will not reach all nodes.

[0003] Directional radios: Directional radios allow the medium to be separated into smaller contention zones (as compared to non-directional antennas). On one hand this can potentially lead to higher throughput if the scheduler can properly utilize the smaller contention regions, but on the other hand it complicates scheduling and collision avoidance, which nominally relies on mechanisms which rely on all nodes that may interfere with a receiver being able to hear a clear to send signal (like 802.11 IBSS mode), or on schedule announcements from a central controller (like 802.11 BSS mode).

[0004] For example, consider Figure 2 where nodes A and B start to communicate with their antennas aligned. If we were using omni antenna, then only A and B could have communicated and C and D would have sensed the medium to be busy and stay idle. With directional antennas one could hope to do better. However with directional antennas C cannot sense that the medium near B may be busy and send to B interfering with its reception of packets from A.

[0005] The method proposed does not require scheduling signals to be duplicated – each node signals each of its neighbors exclusively, thereby exploiting directional nature of the antenna.

[0006] Independent clocks and synchronization: Creating a schedule will however not guarantee that it is followed. It is a well known problem of distributed scheduling that the notion of time in different nodes, even if they have the same hardware, is non-uniform and varies with time. So following a pre-determine schedule will fail because of the relative drift of timing references in nodes.

[0007] Some form of periodic synchronization of clocks is required to deal with the drift between nodes. In several networks periodic messages are sent from a designated node as a reference to which others need to synchronize their time, (e.g. beacons in 802.11, heartbeats etc.).

[0008] However in systems with widespread nodes it is clear that no one node can reach all other nodes and convey time at one instant. Even if the nodes were close to each other they may not be able to hear beacon-like message if their antennas are directional. One can attempt to relay a time message, however, the amount and variation of delay in relaying such a message will lead to inaccuracies, in addition to being complex and expensive. Finally, relaying beacon-like messages will likely be unsuccessful due to different degrees of variations where if a node synchronizes with one neighbour by adjusting its clock, it may in fact be de-synchronizing with another neighbour. Many approaches have been attempted to resolve these problems, including the following.

[0009] Centralized scheduling: The 802.11 MAC [1] has its own scheduling algorithms. In the BSS mode of the algorithm, the Hybrid Control Point function in the access point determines the schedule based on the requests sent in by the stations and conveys this back to them. Each station then follows this schedule until the next time they receive another schedule. This method is only useful when all the stations can hear the schedules being broadcast by the AP. In a network using direction radios, such schemes will fail because the broadcasts will not reach nodes that are out of range or are not along the beam containing the transmitted schedules. The method described here however does not depend upon a single controller or upon a broadcast mode of operation. In fact, as long as a new node can communicate with one of the nodes already in the system, the new node can be incorporated and will be a part of the distributed schedule being maintained through the network. In addition complexity of computing schedules at a central point will increase as the number of links and nodes increases. It is also conceivable that the centralized schedule would need input from all the nodes about attributes like data queues, which will likely become inaccurate if the system is dynamic. However, these problems are seen to be alleviated by distributed scheduling (e.g. the closer decisions are made to the node the less error in estimating queue sizes and fewer links to deal with).

[0010] Decentralized scheduling: In 802.11 IBSS mode and in the Distributed Control Function mode of a BSS, each node independently attempts to send data

after sensing its channel. This strategy allows networks with widely separated nodes as there is no requirement for the broadcast of a central schedule. As the number of nodes and traffic increases on the medium, nodes that sense the medium to be busy will perform exponential back-off required for such random access of a shared medium and thus the throughput of the system will be much lower when compared to the method proposed here.

[0011] Arbitration: Arbitration schemes that attempt to provide coordination such that two neighbors can exchange data when they have packets in their queues, such as RTS/CTS signaling may alleviate some of the overhead of exponential back-off. RTS/CTS schemes have been adapted to directional antennas [2][3], where parts or all of the signaling is done using directional antenna in order to try to encourage multiple simultaneous non-interfering transmissions [4]. However arbitration schemes like RTS/CTS incur costs that grow with the number of nodes in the network and setting up synchronization when a data packet to be sent out on a link arrives, may take an arbitrary amount of time depending upon the traffic serviced by the neighbors. The method proposed here is constant cost (independent of the number of nodes and their distribution) and provides bounded response time and does not require any back-off. In addition the proposed mechanism exploits the separation of media provided by directional antenna (and multiplexing other parameters) to obviate the RTS/CTS synchronization with a schedule that is guaranteed to be such that at no time sending receiving pair non-distinct. Given properly separated media this property of the schedule allows for interference less function without RTS/CTS or back-off overhead.

[0012] Hierarchical: A master-slave relation like the BSS mode can be extended such that master's slave can be a master to other slaves, thereby forming a hierarchical tree of nodes [5]. Such a system can prevent back-off because schedules for each node will be determined by its corresponding master. However, the drawback of such a system is that the root of the whole tree or roots of sub-trees become points of failure, making all the subtending nodes vulnerable. In addition, propagating the schedule from the root to the

leaves is likely to be cumbersome and susceptible to clock errors, where the time reference in all the nodes will need to be guaranteed to be within bounds for the schedule to work. Otherwise if a node's time reference deviates, it will, for instance, start sending data when it's not supposed to and disrupt the system. The method described here has no masters that define the schedule for slaves. Indeed each node is independent to infer radio parameters to use from its environment and create its own schedule. This way, failure in the schedule of a node only affects its own schedule and the interaction with its neighbors, while interactions of the neighbors with their other peers are not affected. In addition, two peers keep the notion of their relative time within bounds, which incurs much less overhead than all nodes synchronizing to a wall clock determined by the root of the tree.

[0013] The method described here does not require any node to ever change its own notion of time, hence avoiding de-synchronization with other nodes as described above. It relies on relative time intervals, using an arbitrary and coarse-grained notion of clock frequency ("ticks per second"), and attempts to bound the variation in such interval with each of its neighbors independently of others by periodically redetermining a fresh set of intervals, thereby avoiding drift.

Summary of the Invention

[0014] The present invention shows a non-interfering schedule of an ad hoc network of nodes with parameterizable radio, which can be achieved by nodes that independently coordinate to form time varying communicating pairs (that are guaranteed to be unique) by local message exchange. The present invention as described herein is a novel distributed algorithm that creates local schedules in such a way that the local schedule in each node and its corresponding data transmissions over the shared medium are non-interfering with the schedules of other nodes and their transmissions. The algorithm works by isolating pairs of nodes as communicating units and maintaining uniqueness of such pairs in time. This along with parameterized radio ensures that non-interfering node pairs in proximity as well as throughout the network can use the shared medium to

communicate at the same time without interfering with each other, obviate previously proposed collision avoidance schemes.

[0015] The novelty of this method is the creation of node pairs, which are isolated using parameterized radio to simultaneously communicate over the shared medium and therefore do not need any other access control. In this way, it is expected that the nodes will be able to achieve high parallelism in data exchange, as well as maintaining robust and scalable scheduling.

[0016] The basic principles of the method proposed here include:

- whether a link is to be active at a time or not is determined at the node pair associated with it depending upon locally available free time slots
- how long a link should exist is preferably determined at the node pair associated with it depending upon local data queues
- the method maximizes utilization of the medium by maximizing the number of node pairs that transfer data to each other without interference by creating a suitable schedules while exploiting parameter diversity in the pairs

[0017] Communication at any time can be viewed as bits traveling over a link from one node to another. A link therefore defines a node pair. In wired networks links are generally constant over time and their permutations are not of interest. But in the case of a shared radio network the links and therefore the node pairs, vary in time and the way they are created determines the nature of the network and medium utilization. In fact node pairs are naturally formed in shared radio networks, in that the basic act of discovering a wireless node forms a node pair.

[0018] The method proposed creates and uses a completely distributed schedule. Each node runs a local scheduler that determines the timing and sequence of its local activities. The collection of all the local schedules forms the global network schedule, which need not be computed as a whole. Intead of creating the schedule at a point and then distributing it to nodes as done in Hybrid control Point, or indeed sub schedules that can be passed down from master nodes to slave nodes, each node's scheduler is solely responsible for itw

own schedule. No element of a node's schedule is computed anywhere but the node. The information required to determine the schedule is local to the node or its direct neighbour and is therefore likely to be accurate even in dynamic systems.

[0019] In general the communication required for determining schedules such that a link between two nodes stays up is determination of exchange of the next time two neighbours are to communicate and for how long, repeated over time. Indeed with this atomic exchange each node can independently construct its own schedule. This is also useful since nodes need not communicate with far away nodes, in fact the scheduling messages need not even be relayed. In this way the schedule created avoids the errors associated with stale information and errors in conveying schedules that a centralized scheduler would encounter. In addition failure of any node in a system affects just that node and none other. Note that time can be viewed as another means to multiplex the radio along with other radio parameters.

[0020] Those skilled in the art will appreciate the scope of the present invention and realize additional aspects thereof after reading the following detailed description of the preferred embodiments in association with the accompanying drawing figures.

<u>Breif Description of the Figures</u>

Figure1 provides a network of nodes having shared radios in accordance with an embodiment of the invention

Figure2 depicts interference associated with directional antennas in accordance with an embodiment of the invention

Figure 3 provides a message negotiation in accordance with an embodiment of the invention

Figure 4 shows drift in nodes in accordance with an embodiment of the invention

Detailed Description of the Preferred Embodiments

[0021] The embodiments set forth below represent the necessary information to enable those skilled in the art to practice the invention and illustrate the best mode of practicing the invention. Upon reading the following description in light of the accompanying drawing figures, those skilled in the art will understand the concepts of the invention and will recognize applications of these concepts not particularly addressed herein. It should be understood that these concepts and applications fall within the scope of the disclosure.

[0022] Pairwise independent scheduling exploits radios that are parameterizable to be non-interfering (e.g. operate in different frequencies and/or use beamforming antenna to localize their area of interference) to provide a robust, scalable (in the number of nodes and distance) and adaptive distributed scheduling mechanism at a low, fixed overhead cost of periodic coordination between neighboring node pairs.

Local scheduling to achieve robust inexpensive global scheduling

[0023] Instead of creating a schedule that encompasses all nodes at once (e.g. a global schedule), or a cascading schedule determined by a local master for its slaves, the method described here decouples the network's schedule into independent schedules for neighboring node pairs. Each node is responsible for its own schedule. The synchronization is between two neighbors and is independent of their synchronization with other neighbors. This allows nodes to form pairs that can communicate at the same time without collisions by exploiting the separation of the shared media afforded parameterizable radio. The schedule created is guaranteed to be such that at any time each sender-receiver pair is globally distinct in its constituent nodes from all other pairs, and by choosing suitable parameters for its radio each pair is made locally non-colliding (the same parameters can be used by other nodes that may be far away). The exact algorithm to choose such parameters is not described here.

Localized contention regions

[0024] The method described herein creates a schedule that is guaranteed to be such that at any time each sender receiver pair is globally distinct in its constituent nodes from all other pairs. In other words at any time no node will attempt to send to a node that some other node is already sending to. Therefore, there is no chance of collisions where more than one node sends to the same receiver. In addition the shared media is divided into localized non-interfering contention regions by choosing parameters for its radios such that no other nearby node is interfered with (however the same parameters can be used by other nodes that may be far away). This prevents collisions at a receiver due to a sender that did not intend it as a recipient. Together, the schedule and parameterized radio therefore provide a network in which each pair operates in its contention-free regions that are separated spatially, temporally or by radio parameters.

<u>Independence from shared clocks purely interval based scheduling</u>

[0025] Pairwise independence in scheduling allows each node to maintain its own clock independently i.e., it doesn't need to change it's own clock to match the clock of any other node (e.g. master). The schedule of communication between two nodes works on time intervals, not external time or wall clocks. The start time and duration of the intervals are negotiated between the neighbors as required. They then independently count down to the event and if their clocks are within error bounds useful information exchange can occur. Note that this interval is such that two neighbors can synchronize to without altering their wall clocks. Each node also ensures that its previous commitments are not changed (due to any external synchronization or reset of wall clocks,) thereby atomically ensuring that subsequent commitments do not conflict with previous ones, thus providing non-preemptive scheduling.

[0026] As an example of pairwise independent scheduling in the context of Figure 1, all nodes would be able to figure out their own schedules independently and the signaling will be such that each node only signal its immediate neighbour (e.g. H need only signal G – the node nearest to it). Signalling and sending data

to the nearest neighbours also minimizes interference and allows for better sharing of the medium. For example in Figure 1, if H signals to C then its unlikely that the link G<-> D can function at the same time since G will experience interference.

[0027] With directional antennas existing medium arbitration methods do not work, so the schedule is created in a way that pairs are always guaranteed to be unique. That is at any time if the schedule in a node pair indicates that they are to communicate then no other node's schedule will have an entry for communicating with either node in the pair. For example in Figure 1, the method proposed creates a schedules where C<->B collisions can never happen and the pairs of A<->B can communicate at the same time as the pair C<->D without interfering with each other. Such multiplexing yields even more benefits in larger networks as seen in the previous figure where many pairs (e.g. A<->C, F<->I, D<->E, G<->H, K<->B) can exist as opposed to much fewer such pairs if proper scheduling was not done or if omni antennae were used. In addition signaling is only between node pairs such that omni directional broadcasts or related signals are not required.

[0028] We use the following model to define the problem setting. There exists a set of nodes in the network $N = \{n_0, n_1, ..., n_k\}$. Each node n_i can determine a set of it's neighbors, i.e. nodes that are first degree peers $N_{Bi} = \{n_p, n_q, ..\}$ that it can communicate directly with. Upon initialization a node will attempt to gather information about its peers and try to communicate with them in order to create its neighbor set. The specifics of such a method to discover neighbors is beyond scope. We assume that after a node pair has established themselves as neighbors (i.e. added themselves to the others neighbor set) the two will be able to communicate for some time sufficient to exchange a small constant number of messages.

[0029] Time in a node is a non-decreasing counter, which can be read at any instant in a variable T. This counter can be viewed as slotted, where a time slot is an atomic time unit for the scheduler (i.e. a singular useful function can be accomplished in a time slot). A node's schedule is then a set of contiguous slots

$SC = \{S_0, S_1, S_2, ..., S_k\} = \{t_0, t_1, t_2 ..., \}$, where t_0 is the start of the next slot as well as the end of the present slot; t_1 is the end of slot 0 and start of slot 1 and so on and slot S_i occurs before S_j implies that the corresponding start time values are such that $t_i < t_j$. Furthermore the schedule only has entries for the finite future (i.e., t_k is a finite value of the time counter) and the current value of T is never greater than *t0*.

[0030] Each node creates a schedule similar to the one above, such that, if in a node n_i the schedule for any time instant indicates that it is to communicate with a neighbor n_j then there is a corresponding entry for the same time instant in the schedule of n_j that indicates that it is to communicate with n_i. This is called the *exclusive pair condition.* If time was denoted in the schedule in terms of contiguous discrete intervals, called *slots,* then the exclusive pair condition implies that if in a node n_i, the schedule for a time slot indicates that it is to communicate with a neighbor n_j then there is a corresponding entry for the same time slot in the schedule of n_j that indicates that it is to communicate with *ni.* Such time slots are represented as $S_i = \{t_i, n_i, p_i, c_i\}$, where t_i is its start time i.e., the value of T at which slot *I* is to start, and n_i is the corresponding neighbor, p_i a tuple of radio parameters that is to be used for the slot and c_i a list of nodes that had contended or may contend on this slot. Note that $n_i, p_i, c_i = null$ for the times when there are no neighbors to communicate with.

[0031] When a new node starts, its scheduler will have slots with no neighbors. To begin with discovery of other nodes need not be coordinated in time and may happen by random overlap of radio parameters and time. However, once discovered neighbours exchange messages and reserve slots in each others schedules, allowing subsequent scheduled negotiations to be coordinated in time. Since the reservations are pairwise, if each node reserves slots in a serialized fashion and ensure non-preemption in its schedule it can satisfy the exclusive pair condition. This implies that the schedule created would have successfully created as many different pairs as possible thereby maximizing medium usage, given that the radio parameters of two links do not overlap in space or time.

Negotiating

[0032] Since the slot will begin a finite amount of time after the exchange to reserve the slot, a function that computes the expected data to be exchanged using the present queue length and history is used at the time of negotiating the slot size. By using suitable policies, such as capping the maximum slot that a neighbour is assigned and each busy neighbour is given equal chance to book slots, nodes can ensure that their schedule is fair. Variations of booking schemes, such as time proportional to the ratio of data expected on one queue to data on all the queues, can be adopted according to the particular application to tailor how the neighbours are allocated time. A generalization of the booking function is as follows:

$B_{pq} = \{ s_i : s_i$ is a slot in the future that is booked for data transfer from node p to $q\}$ and can be computed by the function $\beta_{pq}(.)$ ...(2)

$$B_{pq} = \beta_{pq}(q_{pq}, q_{qp}, F_{pq}, F_{qp}, B'_{pq}, B'_{qp}, Q_{pq}, Q_{qp}, P_{pq})$$

where:

q_{pq} = the queue size for data from node p to q

$F_{pq} = \{ s_i : s_i$ is a slot in the future that is available to be booked for data transfer from node p to $q\}$

$B' = \{ s_i : s_i$ is a slot in the past that was booked for data transfer from node p to $q\}$

$Q_{pq}, = \{q^i_{pq}, : q^i_{pq}$ is the change in the queue size for data from node p to q in the slot $i\}$

P_{pq} = {a tuple of radio parameters available for the slots for data transfer from node p to q}

[0033] A message exchange sequence is used to negotiate the free slots and the time required for the nodes to drain their data. Negotiation between two nodes A and B is as follows. As shown in Figure 3 first A notes its clock value at present as a reference time for the present negotiation and computes the tuple $\{q_{AB}, F_{AB}, B'_{AB}, Q_{AB}, P_{AB}\}$ where the elements of the tuple are defined above. A

ensure that all time values in the message is relative to the reference time. This message is sent to *B* which first computes its own tuple $\{q_{BA}, F_{BA}, B'_{BA}, Q_{BA}, P_{BA}\}$ with its own reference time. Then the booking functions for the directions *AB* and *BA* are computed by matching the free slots of both *A* and *B*, to find common time intervals, *i.e.* slots (or parts of slots), from F_{AB} and F_{BA} until all the free slots are exhausted or time for all q_{AB} and q_{BA} has been booked. Also the argument for radio parameter space is used to pick suitable radio parameters. The locally available parameter space is used to pick suitable radio parameters. The locally available parameter space can be maintained by modules that may scan the environment in idle time. The mechanisms for determining the parameter and setting them are dependent upon the particular radios and are therefore excluded from this discussion. The resulting set of slots, B_{AB} and B_{BA} are said to be the set of booked slots for that negotiation and are conveyed back to node *A* so that it can book those slots in its schedule as well.

[0034] It is possible that there are no parameters that can be used to avoid interference, in which case a node (e.g. in Figure 2) (B) may be able to note the *interfering node* (C) and the upcoming slots that it may likely get interfered within in susceptible slots $I_{(BC)}$, also marks the present link as interfered link. The next time it negotiates with the interfering node it can convey the interfered slot and the susceptible slots as well as the radio parameters it can use in those slots. The interfering nodes can try to switch the radio parameters it can use in those slots. The interfering node can try to switch the radio parameters of its interfering link, given radio parameters of the colliding pair. If this fails, the interfering node marks its interfering links as an interfered link and associated slots as susceptible slots. Unless parameters are available in the future to make the links non-interfering again the markings are retained and new slots are also marked as such. Essentially the pairs now need to share the medium. The interfering node can determine if any collisions are possible by comparing its own susceptible slots with the susceptible slots sent over to see if there is any time when the interfering pairs may be active at the same time. From then on the interfering pair can book slots avoiding the ones that the colliding pair has booked, to be

skewed in time and thus share the link. Figure 3 shows the exchange of collision information and susceptible slots as well as a general avoidance function that works as above.

[0035] Since B_{AB} and B_{BA} are computed out of F_{AB} *and* F_{BA} are only one booking occurs and completes at any given time, the bookings can be guaranteed to be serialized and thus to not conflict with any existing bookings. Note that in order to convey free slots the nodes need to use relative time, because they have free running independent clocks. Relative time is calculated by a common event, referred to as the reference point of the negotiation. The further the reference point of B is from that of A, the more error will occur in the times booked at both ends (note that we use the error term e in calculating the impact of drift as shown in equation (1) – see below). One such reference point could be the reception of *A*'s message at *B* and if the MAC layer provides message acknowledgement the reference points for both *A* and *B* can be very close together in terms of absolute time. Also note that in the above discussion instead of sending the amount of data in queue time measures to drain the queue can be directly communicated and used as arguments to the function (2).

[0036] A given node can repeat this process periodically with all of its neighbours to continue to communicate with them. The message exchange above can be interleaved with user data exchange (e.g. as a preamble, or post-amble). However scheduled communication with a neighbour may be lost if no booking for a future slot exists in the upcoming schedule. In order to prevent this from happening the exchange to future slots books, will in the absence of any data to be exchanged, book small slots that will be purely for the sake of completing another negotiation and thereby keeping the neighbour in the schedule.

Nodes with multiple sharable radios

[0037] Although the treatment of the scheduling problem above assumes that nodes have a single radio to multiplex, the solution proposed above can be easily extended to nodes with more than radios. The problem is easier if the number of

radios is greater than the number of neighbours that a node has to form a link with. If however the number of radios is less than the number of links then the solution above can be applied as such: Each radio will have its own schedule which is maintained independent of the other schedules in the same node, except during negotiations. When the set of available slots it being calculated, slots on all available radios are included. The booking function thus can pick slots of any of the neighbours or its own radio's schedules. If more than one radio is available for a given time the function can be clever to select the previous radio to minimize radio set up overhead. Indeed the equation (2) and its arguments will still hold except that the slots will have a superscript to denote the radio number as s^k_i where k is the radio identifier and i a slot identifier. However an important distinction with the single radio case is that nodes with multiple radios need to ensure the serialization of its negotiations and therefore its slot bookings. This can be easily achieved by ensuring that the negotiation with a peer doesn't start until the time that no other negotiation is ongoing. Alternatively if an implementation is such that this condition is too restrictive then the slots can be guarded by means of semaphores. For instance the negotiation process can mark the slots that it has sent to the other node F_{AB} as under-negotiation and other negotiations will then avoid slots that are marked such. If some of these under-negotiant slots end up being unused they can be marked as free again.

Dealing with clock drift

Furthermore in order to avoid the problem of clock drift the method proposed only used time intervals that can be bounded as a function of the maximum drift in individual clocks. Scheduling proposed here relies on time intervals relative to each neighbour. Given the worst case variation in the clocks the scheduler is able to bound the variation in each interval with each of its neighbours independent of others. By periodically determining a fresh set of intervals it avoids losing synchronization. Indeed the method described here doesn't require any node to ever change its own notion of time, hence avoiding de-synchronization with other nodes. In fact one can compute the maximum time

before two nodes must negotiation another interval. Consider the Figure 4, where a message is sent from B to A. If e is the error in timing on the message exchange and m_u and m_{us} are deemed to be the minimial slot size and the minimal sized slot that can accomplish useful data exchange then:

$$m_{us} \geq (D + m_u) - (e + D + \delta D)$$

$$D \leq (m_{us} - m_u - e) / \delta$$

Where:

D = maximum time before which two nodes must negotiation another interval ≤ D

δ = maximum time that the clocks on two nodes can vary in one unit of time

[0038] The mechanism above allows nodes to schedule data exchange with its neighbors, who will in turn be able to relay data to their neighbors and so forth. In this way, a large network can globally exchange data, hop-by-hop, by using a completely locally determined scheduler, where locality is a node and its direction neighbors. This method is independent of any global clocks and does not require communication with any entity other than direct neighbors. Further this exchange is resilient of clock drifts and avoids interference due to directional antennas.

[0039] Those skilled in the art will recognize improvements and modifications to the preferred embodiments of the present invention. All such improvements and modifications are considered within the scope of the concepts disclosed herein.

<u>References</u>

The following items are incorporated herein by reference in their entireties:

1. IEEE, IEEE Proposed Standard 802.11e (DRAFT), “Medium Access Control (MAC) Enhancements for Quality of Service (QoS)”.

2. Y.B. Ko *et al.*, "Medium Access Control Protocols Using Directional Antennas in Ad Hoc Networks," *IEEE INFOCOM 2000*, March, 2000.

3. A. Nasipuri *et al.*, "A MAC Protocol for Mobile Ad Hoc Networks Using Directional Antennas," *Proceedings of the IEEE WCNC 2000*, Chicago, September, 2000.

4. Y. Wang *et al.*, "Collision Avoidance in Single-Channel Ad Hoc Networks Using Directional Antennas," *Proceedings of the 23rd International Conference on Distributed Computing Systems*," 2003.

5. A. Smith, D. Steer, K. Teo, K. Ng, "ACUMEN: Transit Network Design Asynchronous Design Issues and Solutions," *Livelink*, http://livelink.us.nortel.com/livelink/livelink.exe/5359137/TL_Asynch_Solutions.zip?func=doc.Fetch&nodeid=5359137, December, 2002.

Figure 1

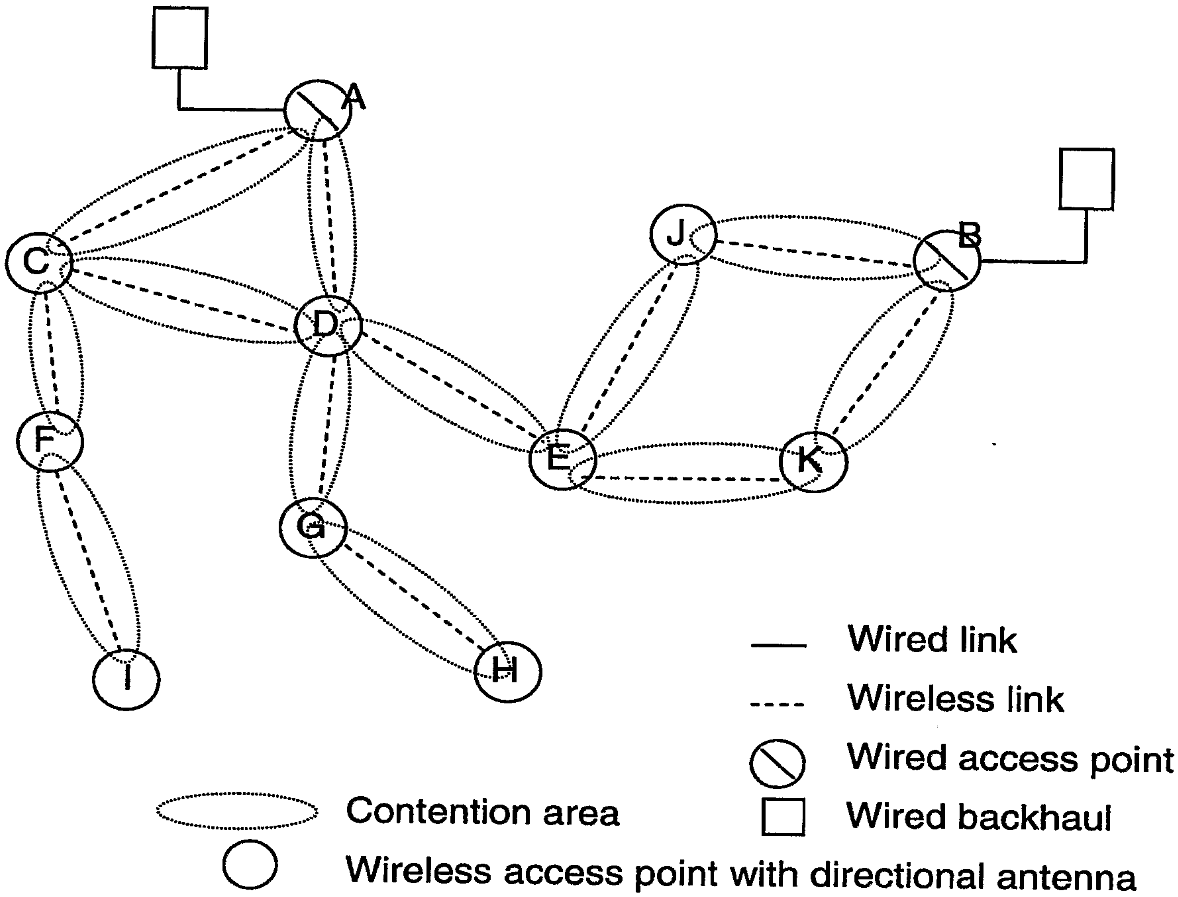

A
B
C
D
E
F
G
H
I
J
K
Wired link
Wireless link
Wired access point
Wired backhaul
Contention area
Wireless access point with directional antenna

Figure 2

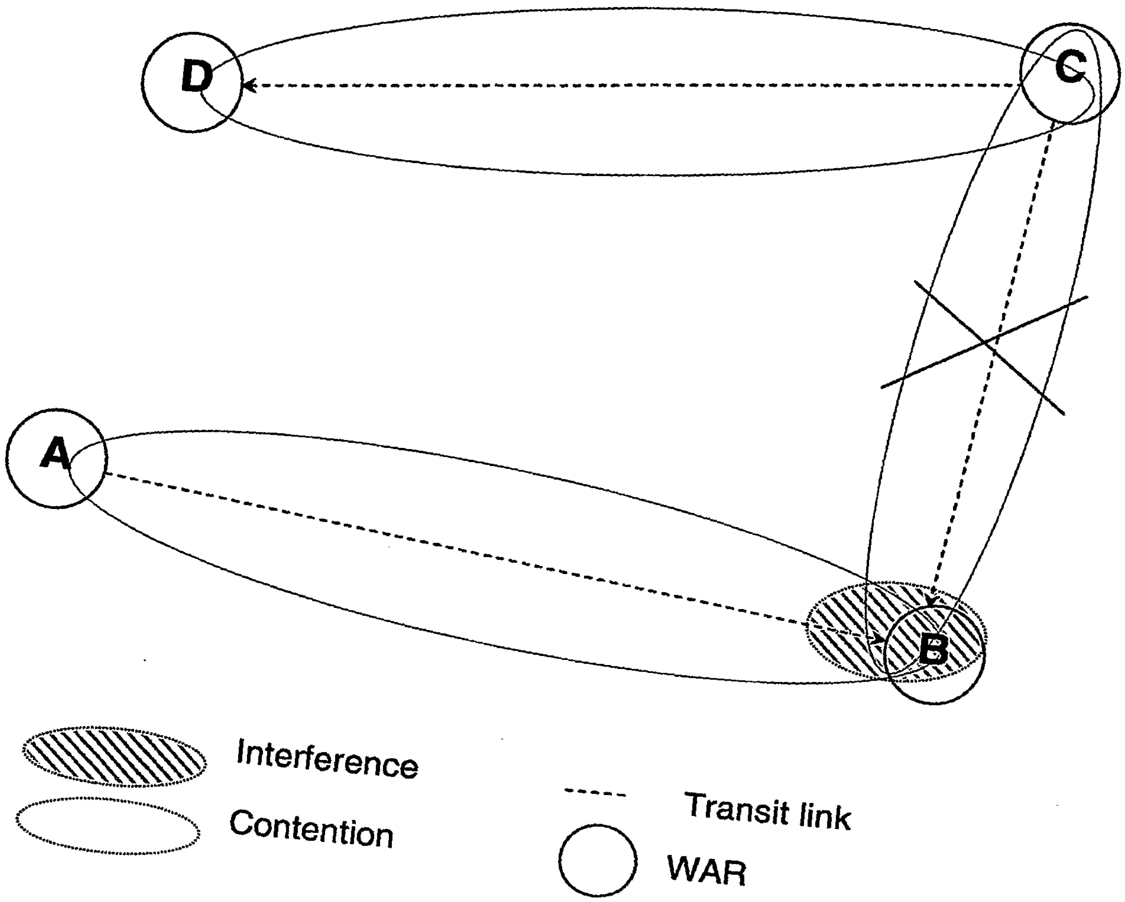
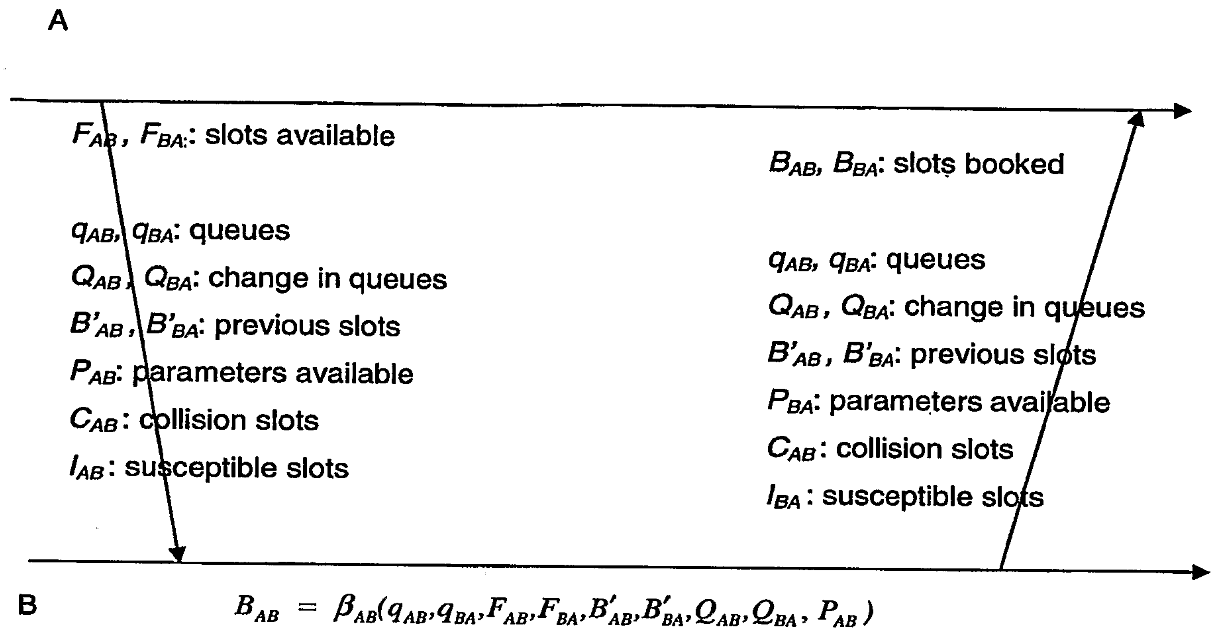


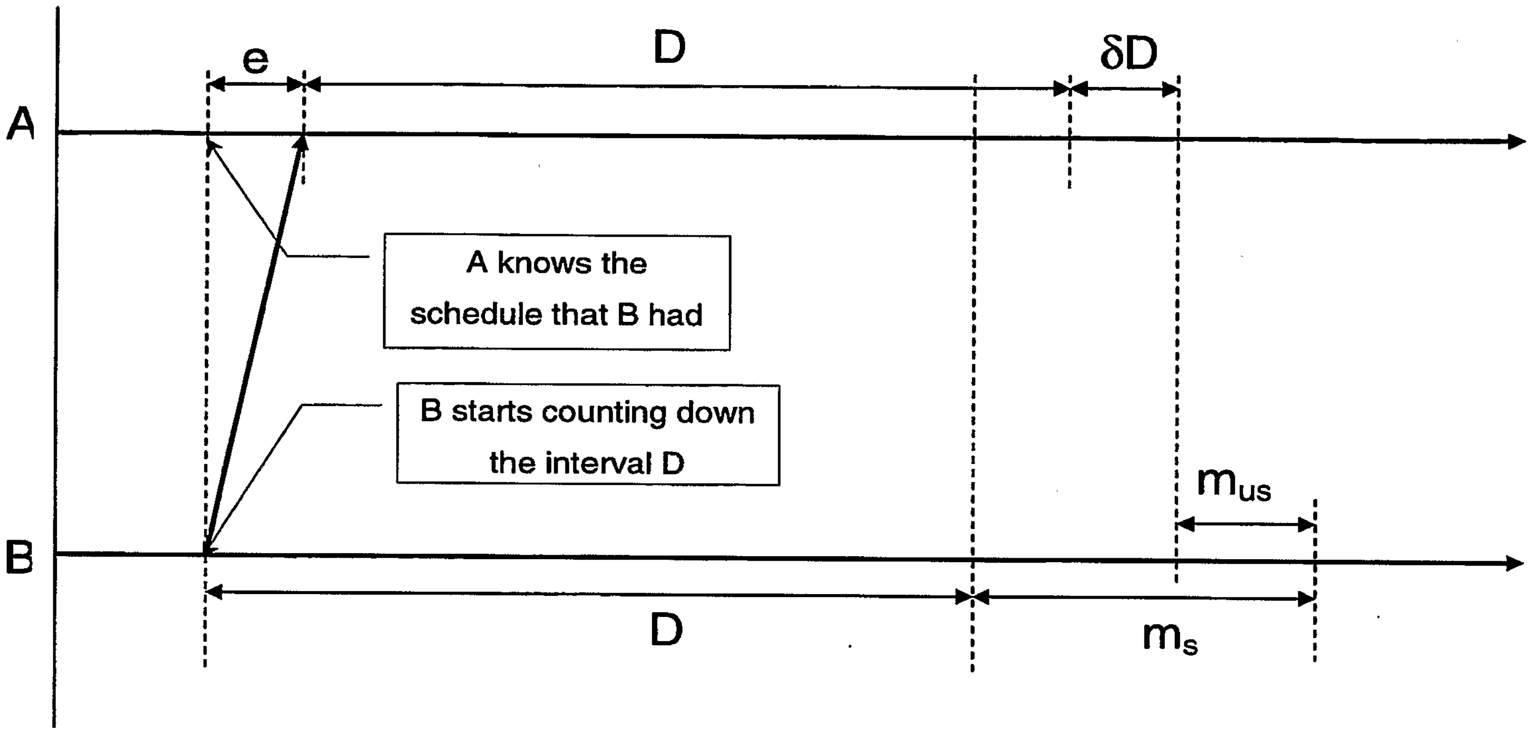
Figure 3



$$B_{AB} = \beta_{AB}(q_{AB}, q_{BA}, F_{AB}, F_{BA}, B'_{AB}, B'_{BA}, Q_{AB}, Q_{BA}, P_{AB})$$
$$B_{BA} = \beta_{BA}(q_{AB}, q_{BA}, F_{AB}, F_{BA}, B'_{AB}, B'_{BA}, Q_{AB}, Q_{BA}, P_{BA})$$

$$I_{BA} = \alpha\,(I_{AB}, C_{AB})$$

Figure 4


e
D
δD
A
A knows the schedule that B had
B starts counting down the interval D
m_{us}
B
D
m_s